A Study of Public Key Cryptography Based Signature Algorithms

***Malabika Das**
Heramba Chandra College, Kolkata, India.
malabika.mail@gamil.com

Rajdeep Chakraborty
Netaji Subhas Engineering College, Kolkata, India.
rajdeep.chak@gmail.com

*****Corresponding author*

Abstract:

Security in the world of the internet has become very important in all aspects of social life. One of the methods of securing the information on the internet is public-key cryptography or asymmetric cryptography. Public key cryptography is not only a process of encrypting information, it also provides confidentiality, data integrity and authentication. RSA and ElGamal are two very important public-key cryptosystems. These two cryptosystems are also used for digital signature scheme because of their high level of security.

In this paper, we discussed RSA algorithm in details with example. RSA is one of the most widely used public-key cryptography in various application. Then we discussed the ElGamal algorithm in details with example. ElGamal public-key cryptography is also used in many applications nowadays. In the next section, we discussed digital signature using RSA algorithm and ElGamal algorithm as digital signature is one of the important application of public-key cryptography. Therefore, Reader will have a good understanding of public-key cryptography. In this paper we have also included a relative study between RSA and ElGamal.

Keywords: Public-key cryptography, Digital signature, RSA, ElGamal

Introduction:

With the increased use of computers and communication systems, information security has become the biggest concern. Cryptography provides for secure communication in the presence of adversaries through encryption. Two forms of encryption are common in use symmetric and asymmetric or public-key encryption. In public-key cryptography [1][2], two different keys are used for encryption and decryption, one is the public key and the other is the private key. Each receiver has its own set of public and private keys. Public keys are kept public and any person can encrypt a message using the intended receiver's public key, but only by the receiver's private key, the message can be decrypted.

Digital signature [1] is a mathematical scheme to verify the authenticity of digital message or document. In this process, the sender attaches a code with the message that acts as a signature. Digital signature is based on asymmetric cryptography or public cryptography. It ensures the source and authenticity of the message. The signer creates the digital signature using a private key to encrypt signature-related data, while the only way to decrypt that data is the signer's public key. Now how it works, when a signer digitally signs a document, a cryptographic hash is generated for the document. The cryptographic hash is then encrypted using the sender's private key which creates the digital signature. Then It is appended to the document and sent to the receiver side along with the signer's public key. The recipient can decrypt the Digital signature with the signer's public key. A cryptographic hash is again generated on the receiver's side from the document. Then both cryptographic hashes are compared to check its authenticity. If they match, the document considered valid. Fig 1 gives the block diagram of Public-key cryptography.

Section 2 describes the RSA cryptosystem with example and used as digital signature, Section 3 describes the ElGamal cryptosystem with example and used as digital signature, Section 4 gives the study in details as literature review. Section 5 draws the conclusion and references are given at last.

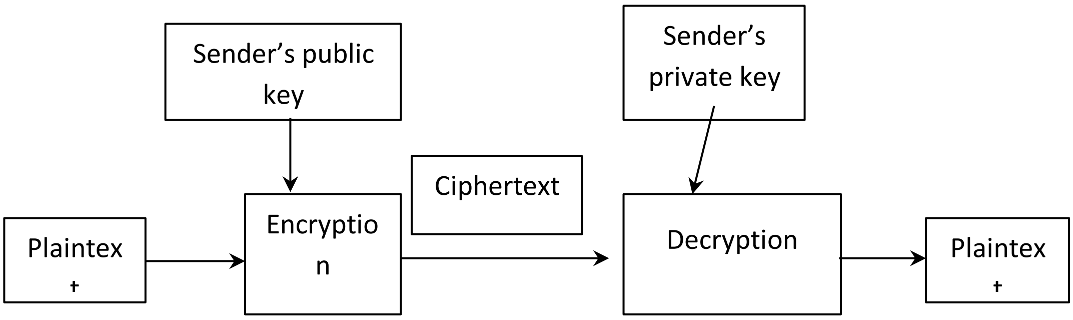

Fig.1: Block diagram of public key cryptography

RSA cryptosystem

RSA cryptosystem [1] is a public key cryptosystem algorithm based on exponentiation in modular arithmetic. The system was invented by Ron Rivest, Adi Shamir, and Len Adleman in 1978 and hence, it is termed as RSA Cryptosystem. RSA Cryptosystem involves three steps: Key generation, Encryption, and Decryption.

Section 2.1 gives key generation, section 2.2 gives encryption method, section 2.3 gives decryption method, section 2.4 gives an example and section 2.5 illustrates digital signature algorithm using RSA.

Key generation:

Each participant needs to generate public and private keys.

- Select two large prime numbers, p, and q.
- Calculate $N=p\times q$, For strong encryption N must be large, a minimum of 1024 bits.
- Calculate totient function $\phi(N)=(p-1)(q-1)$.
- Select an integer e such that e is co-prime to $\phi(N)$, and $1< e< \phi(N)$.
- The pair of (N,e) is RSA public key and made public.
- Calculate d such that $ed \equiv 1 \bmod \phi(N)$.
- The pair (N,d) makes the private key.

Encryption:

Suppose the sender wish to send some text message to someone whose public key is (N, e). The sender then represents the plaintext as a numbers less than N. To encrypt the plaintext P, which is a number modulo N. The Ciphertext C Calculated as-

$$C \equiv P^{e} \text{ mod N.}$$

Decryption:

Using private key (N,d), Plaintext can be found-

$$P \equiv C^{d} \text{ mod N.}$$

Example using small numbers:

Let the plaintext be P=9.

- First, Select two large prime numbers p = 7 and q = 11.
- Calculate N = p × q =7 ×11=77.
- Calculate φ (N) = (p - 1) x (q-1) = (7 - 1) x (11 - 1)=6×10=60.
- Let us now choose relative prime e Such that e is co-prime to φ (N) = 60.Say, e=7.
- Thus the public key is (N, e) = (77, 7).
- A plaintext message P=9 is encrypted using public key (N, e). To find ciphertext from the plain text following formula is used to get ciphertext C. $C \equiv P^{e}$ mod N = 9^{7} mod 77=37.
- The private key is (N,d). To determine the private key, we use the following formula d such that: ed mod φ (N)=1, 7d mod 60 = 1, which gives d = 43.
- The private key is (N,d)= (77,43).
- A ciphertext message C is decrypted using private key (N,d). To calculate plain text P from the ciphertext C following formula is used. P = c^{d} mod N = 37^{43} mod 77= 9

In this example, Plain text P = 9 and the ciphertext C= 37.

Digital siganature with RSA Algorithm:

The RSA public-key cryptosystem is also used to sign and verify messages. Since it is based on the math of the modular exponentiations and discrete logarithms and it's computational difficulty provides a strong security. RSA is an asymmetric digital signature [1] such that one key is used for signing a message and only by the other key the message can be verified.

Where section 2.51 gives key generation, 2.5.2 illustrates signing method and section 2.5.3 illustrates verifying method of digital signature using RSA.

Key generation:

The RSA key-pair consists of:

- public key (N *e*)
- private key (*N*,*d*)

RSA Sign

To Sign a message '*m'* with the private key exponent *d*:

- Compute the message hash: h = hash(m)
- Encrypt h to calculate the signature: s = h^d(mod N).
- The hash h should be in the range [0...N-1). The signature s is in the range [0...N-1).

RSA Verify Signature

Signature s for the message 'm' is verified with the public key exponent e.

- Compute the message hash: h = hash(m)
- Decrypt the signature**:** h′=s^e(mod N)
- Compare h with h' to find whether the signature is valid or not.

If h′=s^e(mod N) = (h^d)^e (mod N) =h, then signature is correct.

Elgamal Cryptosystem:

Elgamal encryption [2][3] is a public key cryptosystem. It is based on the difficulty of finding discrete logarithm in a cyclic group.

Section 3.1 gives key generation, section 3.2 gives encryption method, section 3.3 gives decryption method, section 3.4 gives an example and section 3.5 illustrates digital signature algorithm using RSA.

Key generation:
Participant generates the public and private key pair.

- Select a large number p and a generator g of the multiplicative group F_p of integers modulo p.
- Select a random integer b, 1 ≤ b ≤ p-2, and compute g^bmod p.
- Now public key is (p, g, g^b) and private key is b.

Encryption:
To encrypts a message M, the sender represent M as integers in the range {1,.., p-1}.

- Then sender obtain receiver's public key (p,g,g^b).
- Select a random integer k, 1≤ k ≤ p-2.
- Compute c1= g^k mod p, c2=M × $(g$^$b)^k$.
- Send ciphertext C = (c1,c2).

Decryption:

- Use private key b to compute ($c1^{p-1-b}$) mod p. Note $c1^{p-1-b}$ = $c1^{-b}$.
- Recover M by computing $c1^{-b} \times c2\ \ modp$

Example Using Small Numbers:
This is a simple example of ElGamal cryptosystem.
Let Participant A wants to send a message to Participant B. First A needs B's public keys.
Now, B choses a number p=17 (In practical this number is very large).
Then B Selects a random number b=5 and a generator g =6. Then B calculates g^b=7 and makes (17,6,7) to public. And keep b=5 as private key.

A encrypts a message M=13 (in the range of {1,2,…16}). Then A chooses a random number k=10 (1≤ k ≤ 15). He calculates c1=6^{10} (mod17) =15. He encrypts c2 =13× 7^{10}=9 and sends (15,9) to B.

B receives (15 9) from A.
B's public key is (17,6,7) and private key is b=5,
B now decrypts the message by using the private key.
Decryption : M= 9× 15^{11} (mod 17)=13 [15^{17-1-5} =15^{11}]

B now decrypted the message and received 13, which is the original message.

Digital Signature with ElGamal Algoirthm:

The ElGamal signature [6] scheme was described by Taher ElGamal in 1985.It involves following steps-

Where section 3.5.1 illustrates system parameters, section 3.5.2 illustrates key generation, section 3.5.3 illustrates signature generation and section 3.5.4 illustrates verification.

Sytem parameters

- Let H be a collision resistant hash function.
- Let p be a large prime such that computing discrete logarithm modulo p is difficult.
- Let g<p be a randomly chosen generator of the multiplicative group of integers modulo p.
- The algorithm parameters are (p,g).these system parameters may be shared between users.

Key generation:

- Choose randomly a select key x with 1<x<p-1.
- Compute y=g^x mod p.
- The public key is (p,g,y).
- The secret key is x.
- These steps are performed by the signer.

Signature generation:

- To sign a message the signers performs the following steps.
- Choose a random k such that 0<k<p-1 and gcd (k,p-1)=1.
- Compute r = g^kmod p.
- Compute s = (H(m)-xr) k^{-1}(mod p-1)
- If s=0, start over again.

Then the pair (r,s) is the digital signature of m. The signer repeats these steps for every signature.

Verification:

- A signature (r,s) of a message m is verified as follows.
- 0<r<p and 0<s<p-1.
- $g^{H(m)}$=y^r r^s(mod p).

The verifier accepts a signature if all conditions are satisfied and rejects it otherwise.

Literature Review and Relative Study

Section 1 gives analysis of RSA cryptosystem, section 4.2 gives disadvantages of RSA algorithm, and section 4.3 shows attack on RSA cryptosystem.

Section 4 gives analysis of ElGamal cryptosystem, section 4.5 gives disadvantages of ELGamal algorithm, and section 4.6 shows comparison between RSA and Elgamal.

Analysis Of RSA Cryptosystem:

RSA algorithm is the first algorithm which can be used both for data encryption and digital signatures. It's security depends on the difficulty of decomposition of large prime numbers. It uses Public-key Cryptosystem that have two different keys, called the key pair for the encryption and decryption. It is estimated that the difficulty of guessing the plaintext from single key and the ciphertext equals to the decomposition of product of two large prime numbers.

Disadvantages of RSA algorithm:

In [4] this paper The author mentions some of the limitations associated with RSA. If any of p,q,e and d is known, then the other values are can be calculated and therefore secrecy is needed. Also, In RSA the message length should be less than bit length, otherwise the algorithm may fail. Another limitation of RSA is that it is much slower than other symmetric cryptosystems since it uses the public key. Also, in RSA the length of plain text that can be encrypted is required within the size of N=p*q.

Attacks on RSA Cryptosystem:

In [13] this paper author discussed some of the attacks on RSA

1. Factoring RSA Modulus: Factoring the public modulus is the most evident way to attack RSA cryptosystem. It is assumed that By 2020 1024 bits number will be factored and will not be secured. As a result 2024 bit key should be more secured.
2. Timing Attacks: It has been observed that the RSA algorithm takes different amount of time to perform its crypto operations according to the key's value, so based on the time required to apply the private key to some information, some estimate can be made of the private key.
3. Chosen Cipher text Attack: In this attacker is able to find out plain text based on cipher text using to extended Euclidean Algorithm.

Analysis of ElGamal Algorithm

ElGamal encryption scheme is based on the difficulty of finding discrete logarithm in a cyclic group. One of the strength of ElGamal is its non-deterministic encrypting the same plaintext multiple times will result in different ciphertext, since a random k is chosen each time.

In [15] this paper points out that ElGamal is used in the free GNU privacy Guard Soft-Ware, and other cryptosystems.

Disadvantages of Elgamal

In [15] this paper the author point out that the main disadvantages of El-Gamal is its need for randomness and its slower speed especially for signing. Another disadvantage is that the message expansion by a factor of two takes place during encryption that means the ciphertext is twice as long as the plain text.

Comparison Between RSA and ElGamal cryptosystem

RSA and ElGamal Both are implementation of Public-key cryptosystem. The strength of this algorithm lies in the bit length used. The difficulty level in RSA lies in the factorization of large primes whereas in ElGamal lies in the calculation of discrete logarithms. RSA is a deterministic algorithm while ElGamal is a probabilistic algorithm.

The [6] author in this paper compared the two algorithms RSA and ElGamal. After testing it is concluded that RSA and ElGamal took the same time in the key generation. Though it takes longer time to generate 2048 bit keys as the calculation result have modular expression. It's also concluded that the encryption and decryption time of RSA algorithm is better than ElGamal algorithm. RSA algorithm is faster than ElGamal algorithm. Security wise, The ElGamal algorithm will be more difficult to solve than the RSA algorithm because ElGamal has a complicated calculation to solve discrete logarithms.

Table 1 illustrates the summary between RSA and ElGamal.

Factors	RSA	El-Gamal
Developed	1978	1985
Key-length	>1024 bits	1024 bits
Type of Algorithm	Asymmetric	Asymmetric
Power Consumption	High	low
Key used	Different key for encryption and decryption	Different key for encryption and decryption
Hardware and software implementation	Not very efficient	Faster and efficient

Table 1.Summery Table of RSA and ElGamal:[16]

Conclusion:

In this paper, we have discussed public-key cryptography, which is based on a pair of keys, a public key and a private key. The use of public-key cryptography in digital signature ensures the authenticity and integrity of a message. RSA and ElGamal cryptosystems are implementation of public-key cryptosystem. Both cryptosystems can be used for encryption and signing a message without direct interaction. Also, we have discussed the strengths and weaknesses of RSA and ElGamal algorithms. Finally, through a better understanding of their strengths and weaknesses, further research can be conducted effectively.

Reference

[1] Stalling, W. (2005). *Cryptography and Network Security: Principles and Practices*.India: Pearson.

[2] Meier, A. (2005). The ElGamal Cryptosystem. Retrieved from http://wwwmayr.in.tum.de/konferenzen/Jass05/courses/1/papers/meier_paper.pdf

[3] Grewal, J. (2015). ElGamal:Public-Key Cryptosystem. Retrieved from http://cs.indstate.edu/~jgrewal/steps.pdf

[4] Gupta, S., & Sharma, J. (2012). A hybrid encryption algorithm based on RSA and Diffie-Hellman. *2012 IEEE International Conference on Computational Intelligence and Computing Research,* 1-4.

[5] Mansour, A. H.(2017). Analysis of RSA Digital Signature Key Generation using Strong Prime. *International Journal of Computer (IJC), 24*(1),28-36.

[6] Andysah Putera Utama Siahaan, E Elviwani, and Boni Oktaviana. (2018). Comparative Analysis of RSA and ElGamal Cryptographic Public-key Algorithms. *In Proceedings of the Joint Workshop KO2PI and the 1st International Conference on Advance & Scientific Innovation (ICASI'18). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*), Brussels, BEL

[7]. Haraty, R. A., Otrok, H., & El-Kassar, A. N. (2004, April). A Comparative Study of Elgamal Based Cryptographic Algorithms. *In ICEIS 3*,79-84.

[8] Satake, K. & Kasahara,M.(1997) Fast RSA-Type Cryptosystem with Public Data of small Size. *Electronics and Communication I Japan,80*(2).

[9] Daeri, A. Zerek,A.R. & Abuinjam,M.A.(2014). ElGamla Punlic-key Encryption. *International Conference on Control, Engineering & Information Technology (CEIT'14).*

[10] Zhou, X. & Tang, X. (2011).Research and Implementation of RSA Algorithm for Encryption and Decryption. *The 6th International Forum on Strategic Technology*.

[11] Aryanti A. & Mekongga, I. (2018). Implementation of Rivest Shamir Adleman Algorithm (RSA) and Vigenere Cipher I*n Web Based Information System. E3S Web of Conference* 31, 10007

[12] Jamgekar, S.R & Joshi, G.S.(2013). File Encryption and Decryption Using Secure RSA. *International Journal of Emerging science and Engineering (IJESE),1*(4) .

[13] Al-Kaabi, S.S & Belhaouari, S.B.(2019).A Survey On Enhanced RSA Algorithms.*Computer Science & Information Technology,*123-142,

[14] Shetty, A. Shetty K, S. & K,K.(2014). A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm. *International Journal of Innovative Research in Computer and Communication Engineering,.2(*5) .

[15] Khalaf, E.F &Kadi, M.M. (2017). A Survey of Acess control and Data Encryption for Database Security. JKAU: *Eng.Sci.,28* (1),19-30